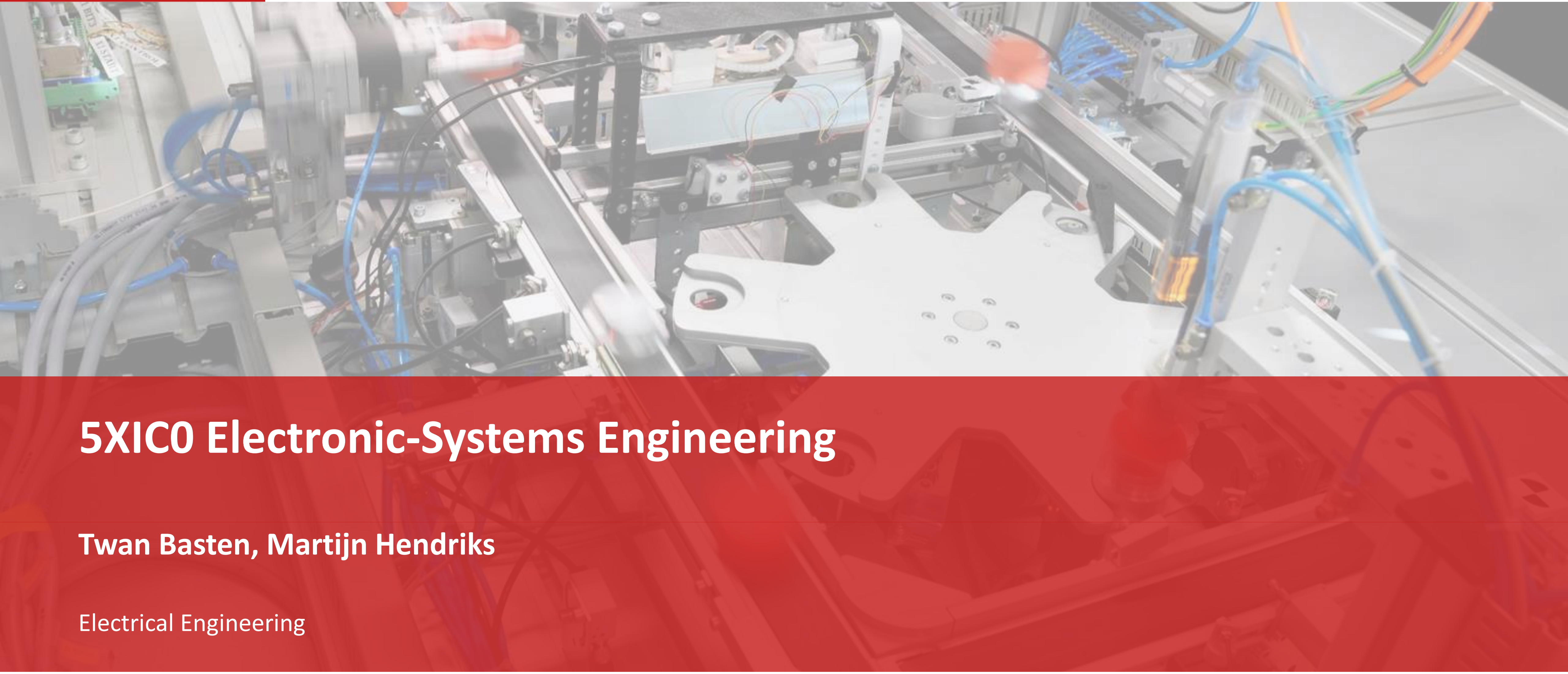
ES ELECTRONIC SYSTEMS

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY



# 5XIC0 Electronic-Systems Engineering

**Twan Basten, Martijn Hendriks**

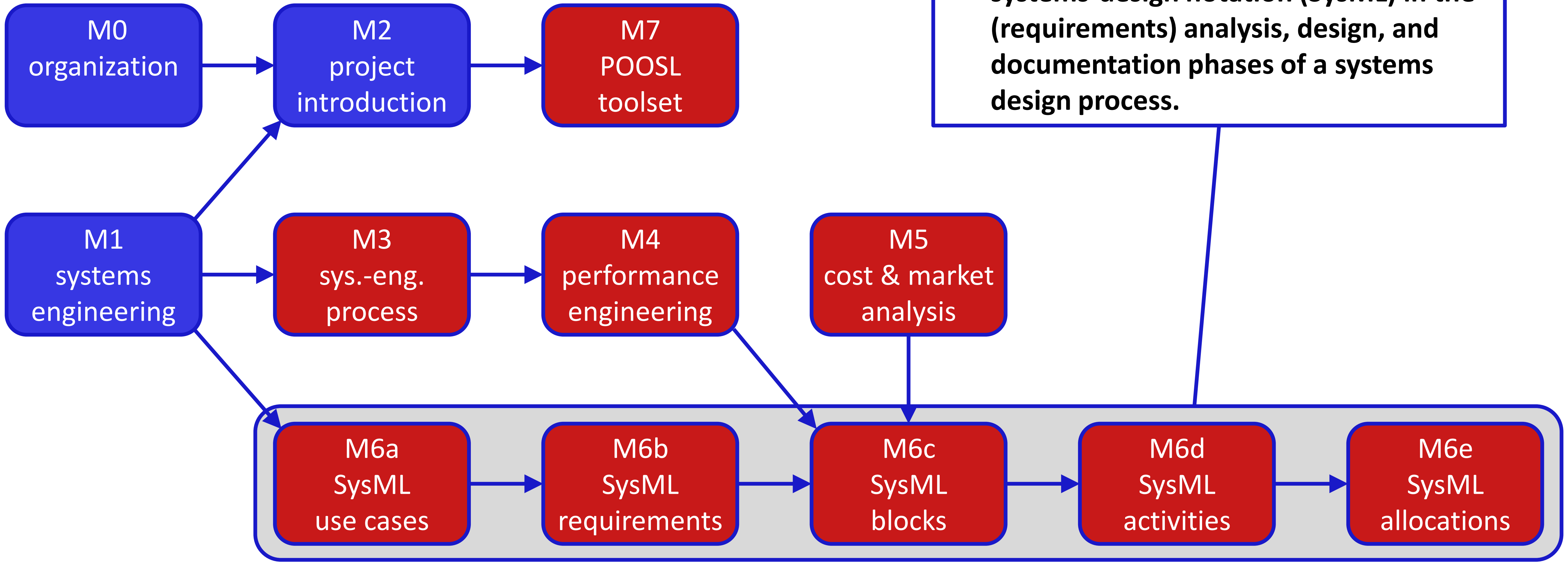
Electrical Engineering

## 5XIC0 Intended Learning Objectives (ILOs)

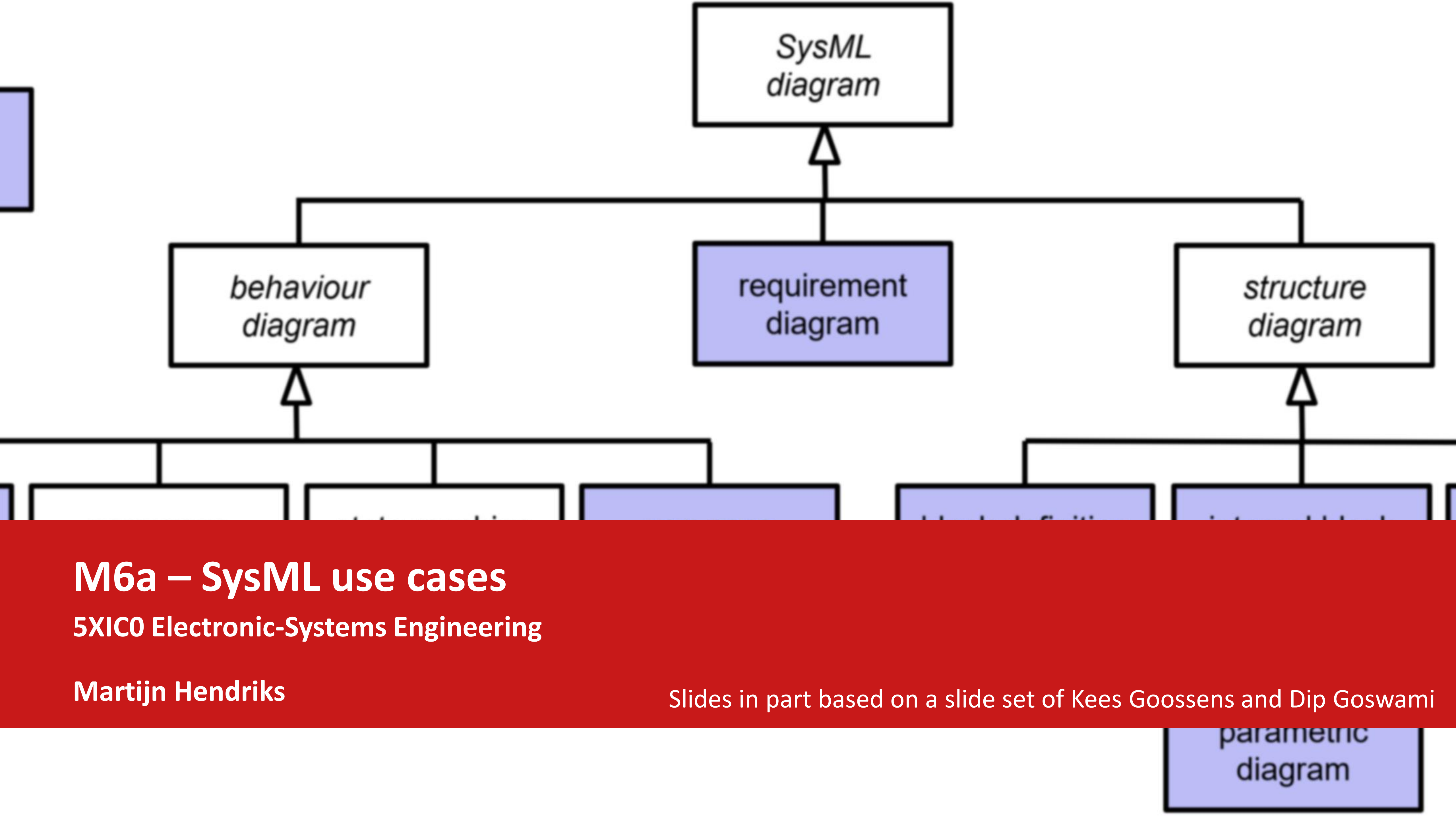
After attending this design-based learning (DBL) course, students should:

1. Be able to successfully apply an electronic-systems design process, as practiced in industry.
2. **Know and be able to use a standard systems-design notation (SysML) in the (requirements) analysis, design, and documentation phases of a systems design process.**
3. Be able to make motivated design decisions taking into account relevant system-level performance indicators and requirements.
4. Be able to select and successfully apply relevant modeling and analysis techniques in support of design decisions.

## modules



2. **Know and be able to use a standard systems-design notation (SysML) in the (requirements) analysis, design, and documentation phases of a systems design process.**



# M6a – SysML use cases

**5XIC0 Electronic-Systems Engineering**

**Martijn Hendriks**

Slides in part based on a slide set of Kees Goossens and Dip Goswami

## in this lecture

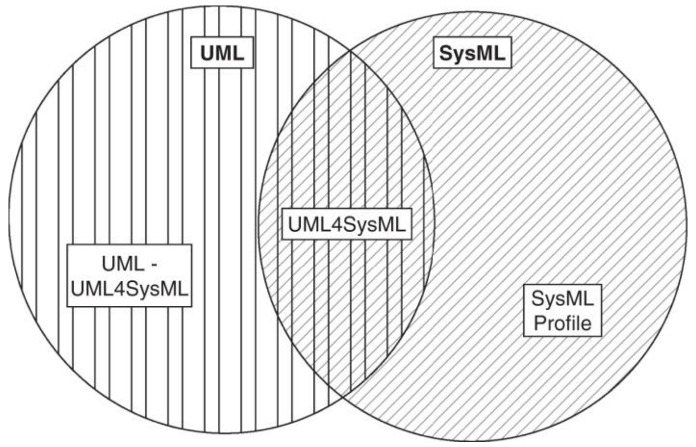
- SysML introduction
- SysML use cases

## SysML – what is it?

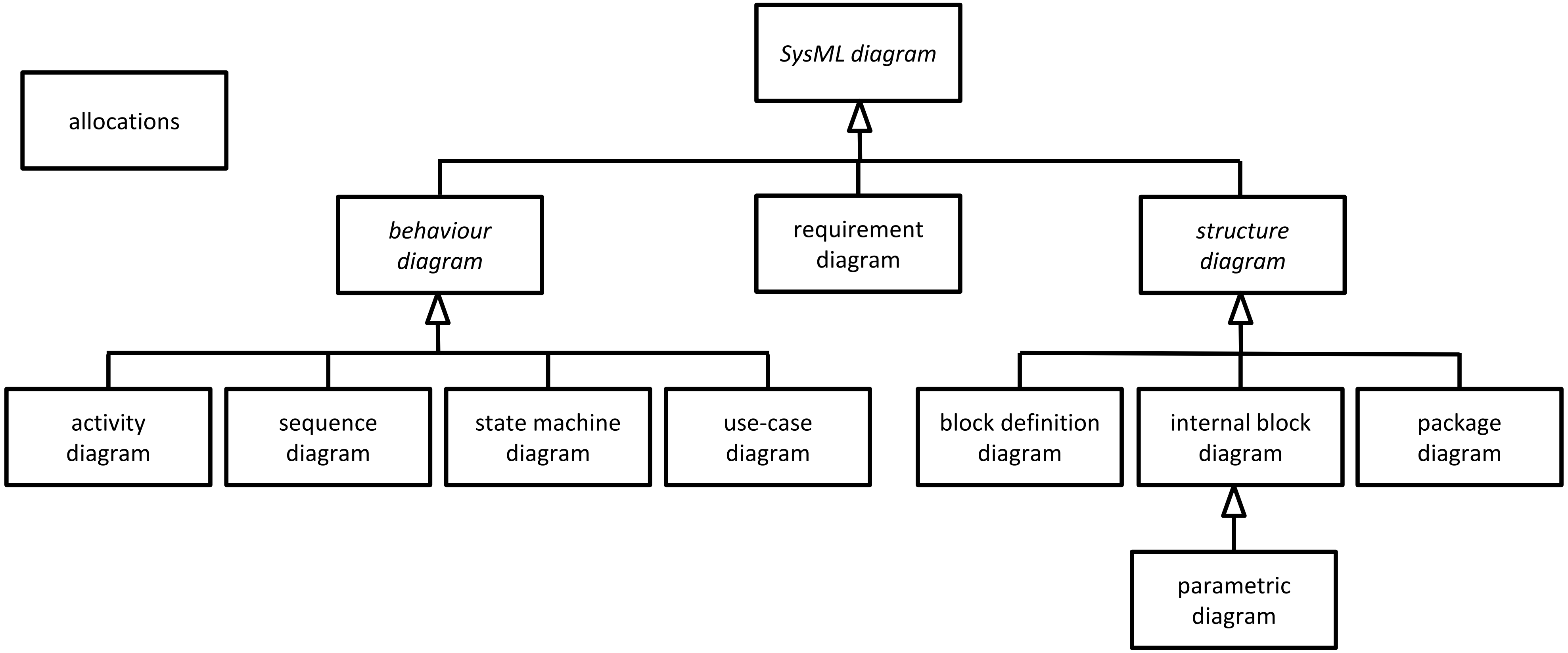
- (semi-)formal graphical modeling language to specify and design systems
  - many types of model elements (structure, behavior, requirements, cross-cutting relations, etc.)
  - a model is a collection of model elements
  - views on parts of the model through diagrams

- standard of the Open Modelling Group (OMG)
  - current version 1.6 (December 2019); 2.0 is on its way
  - https://www.omg.org/spec/SysML/1.6

- SysML is based on the Unified Modelling Language (UML)
  - UML focusses on software development
  - SysML focusses on systems engineering

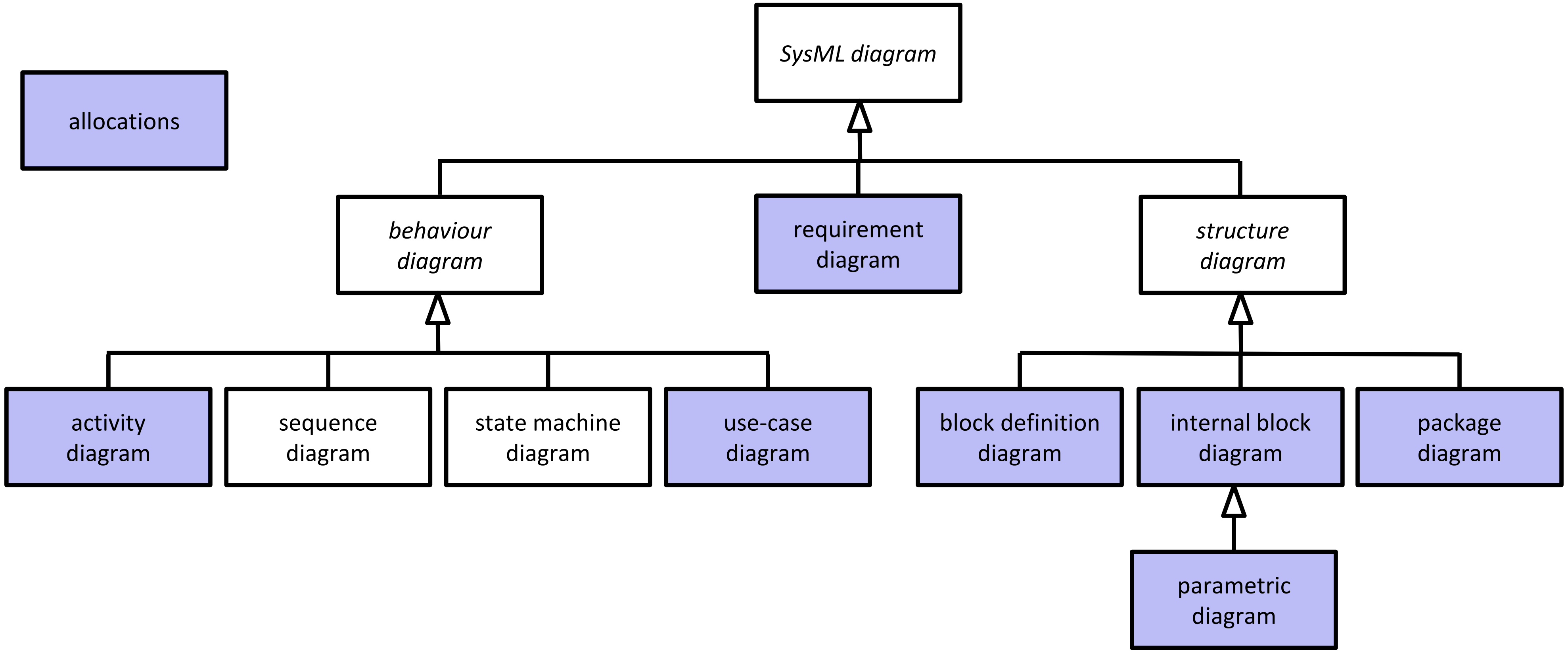


# SysML – diagram overview



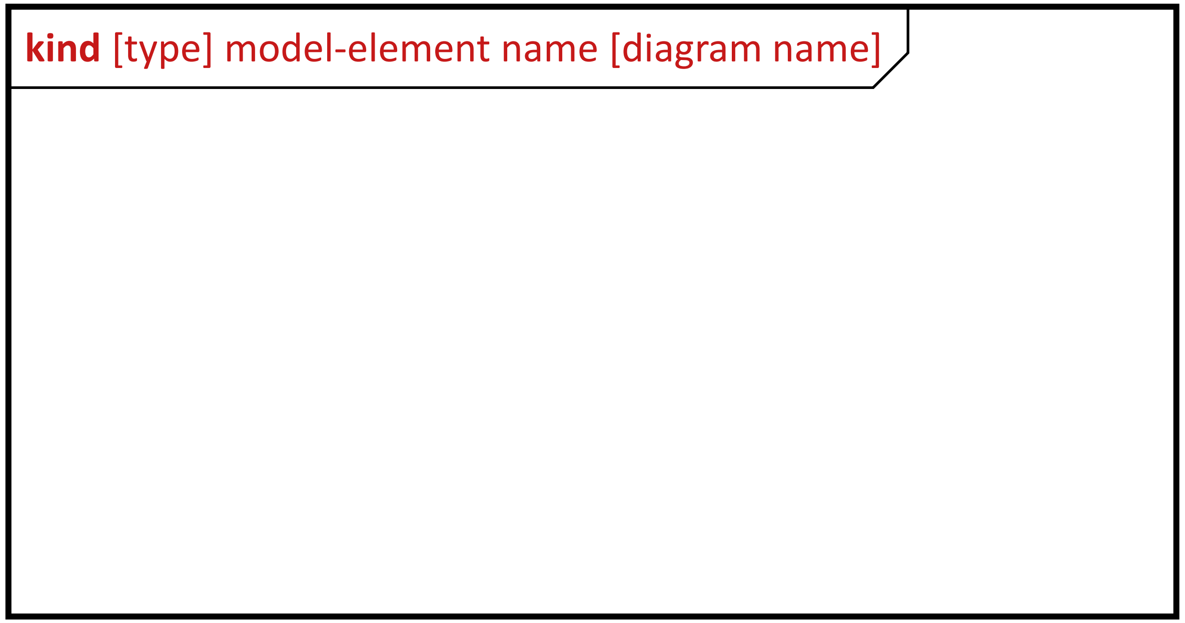
# SysML – diagram overview

diagrams are **views** on the model
(i.e., on a subset of model elements)



# SysML – diagrams

- every SysML diagram has a frame
  - corresponds to a model element that provides the context for the diagram content: the subject
  - model elements are inside frame or on the border
- diagram header consists of
  - diagram kind: the kind of diagram used for the view on the subject
  - model-element type: type of the subject
  - model-element name: name of the subject
  - diagram name: a descriptive name



## SysML – diagram header

- **diagram kind:** act, bdd, ibd, pkg, par, uc, req, sd, stm
- **model-element type** (for a given kind)
  - **act:** activity
  - **bdd:** block, constraint block, package, model, model library
  - **ibd:** block
  - **pkg:** package, model, model library, profile, view
  - **par:** activity, block, constraint block
  - **req:** package, model, model library, requirement
  - **uc:** package, model, model library
  - **seq:** interaction
  - **stm:** state machine

**kind** [type] model-element name [diagram name]

*We will practice this on the go*

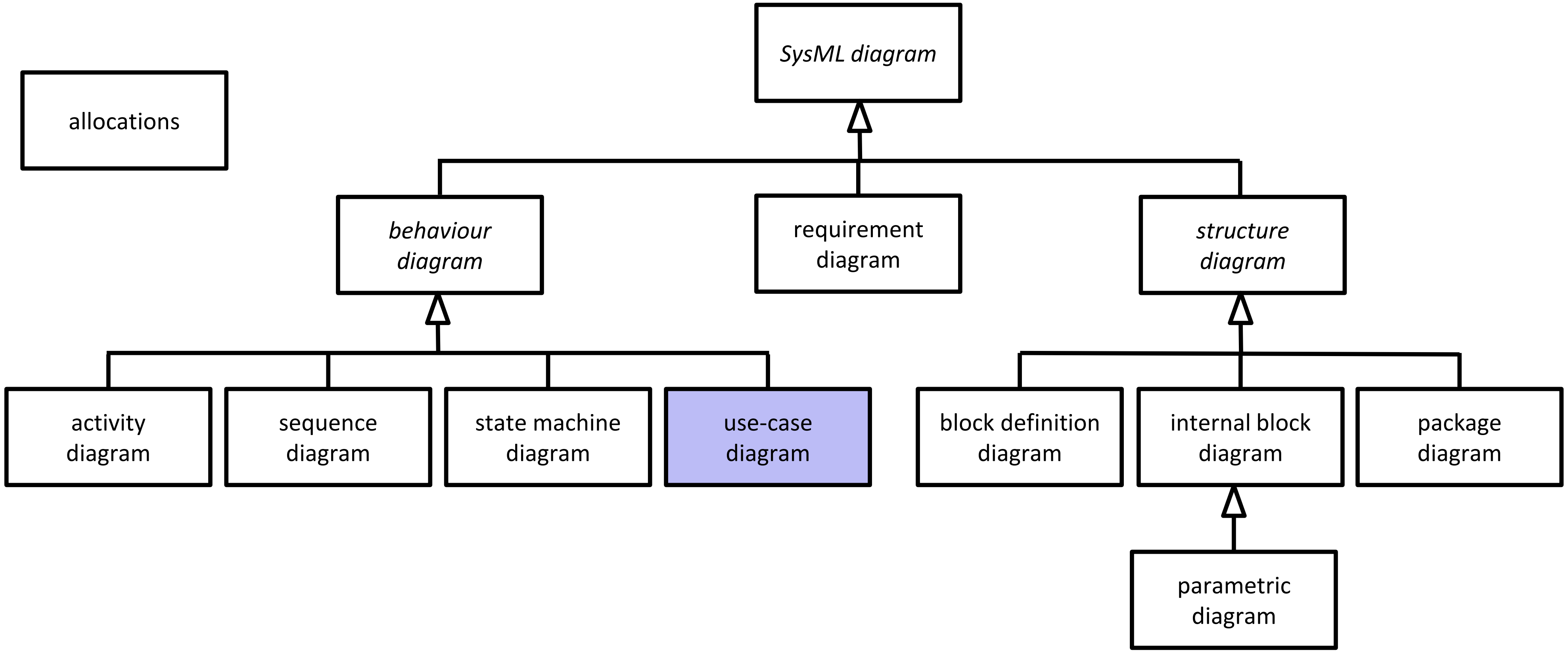
suggested reading: section 5.2

## quiz

- how do a model, model elements and diagrams relate to each other?
    - a model consists of a tree of model elements
    - diagrams are views on the model (and show a subset of model elements) for a specific purpose

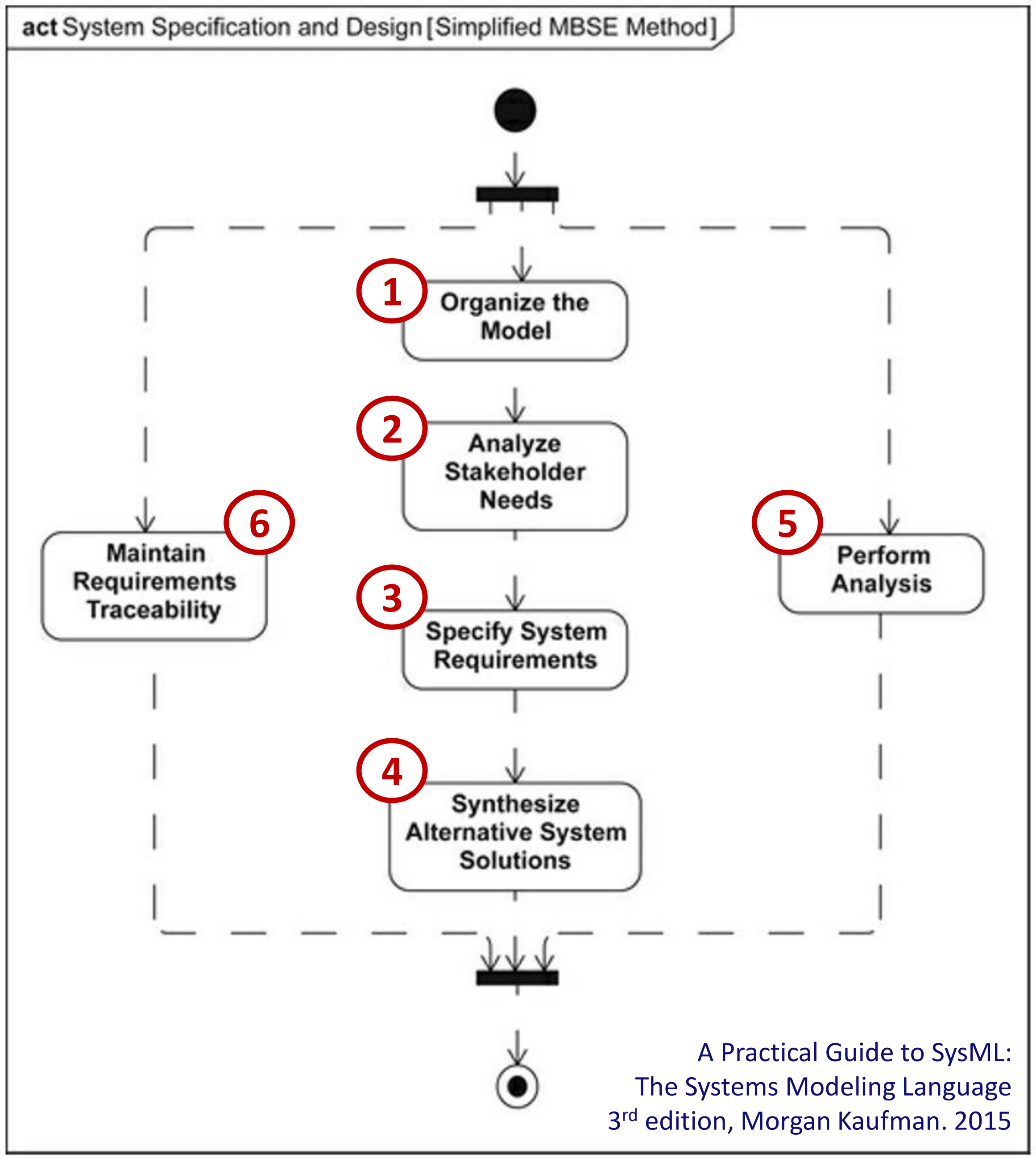
- what are the parts of the diagram header?
    - kind: which kind of diagram (e.g., use case, requirement, block definition, internal block, …)
    - type: the type of the model element that is the subject of the diagram (e.g., package, model, block, …)
    - model element name: name of the subject
    - diagram name: a descriptive name of the diagram

# SysML – use cases



# a simplified² MBSE method

1. SysML package diagram
2. **stakeholders**
   **SysML UC diagrams, UC descriptions**
   measures of effectiveness (moes)
3. SysML requirement diagrams
4. create multiple alternatives
   - SysML BDDs – system decomposition
   - SysML IBDs – interconnections
   - SysML Activity diagrams – UC refinements
   - SysML Allocations – activities to blocks
5. - SysML PAR diagrams – covering all moes
   - POOSL models – makespan
   - verification
6. - SysML allocation – reqs to blocks/activities

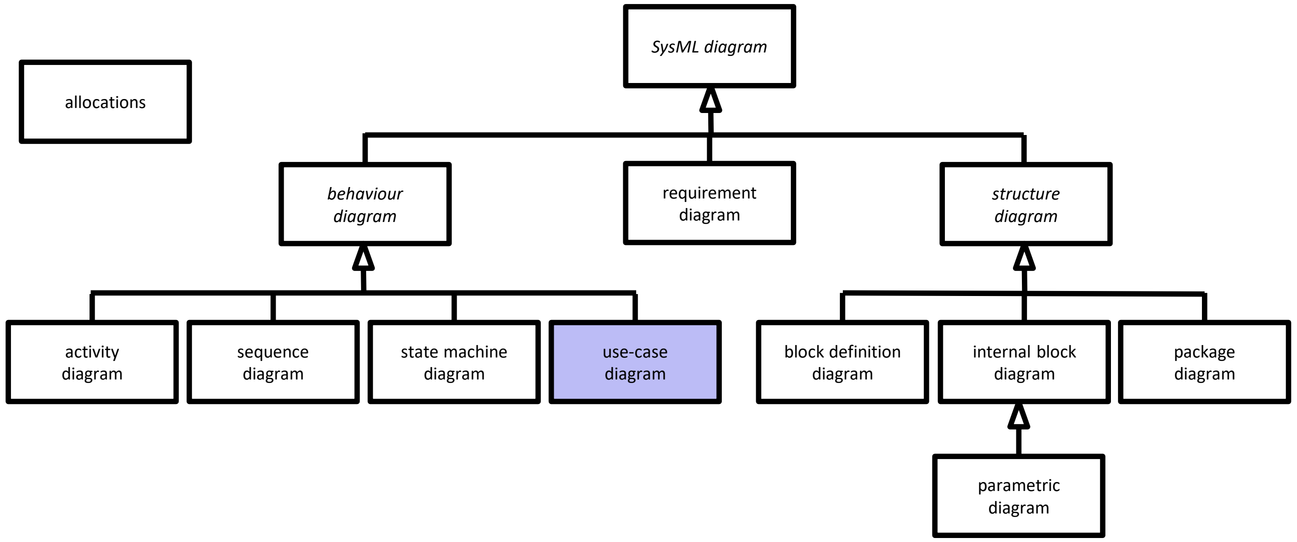

act System Specification and Design [Simplified MBSE Method]
1
Organize the Model
2
Analyze Stakeholder Needs
3
Specify System Requirements
4
Synthesize Alternative System Solutions
5
Perform Analysis
6
Maintain Requirements Traceability

A Practical Guide to SysML: The Systems Modeling Language 3rd edition, Morgan Kaufman. 2015

# SysML – use cases – what & when

method will be introduced in M3 (next lecture)

use cases are used very early in the architecture and design to capture top-level goals the system is intended to support



suggested reading: sections 12.1 – 12.4 (not the part about use-case extension)

# SysML – use cases – model elements

model elements

- subject: system under consideration (a block; use cases are inside)
- use case: functionality of a system in terms of how it is used by its users
- actor: users of a system (e.g., human operators, subsystems in the environment, …)

actor relationships

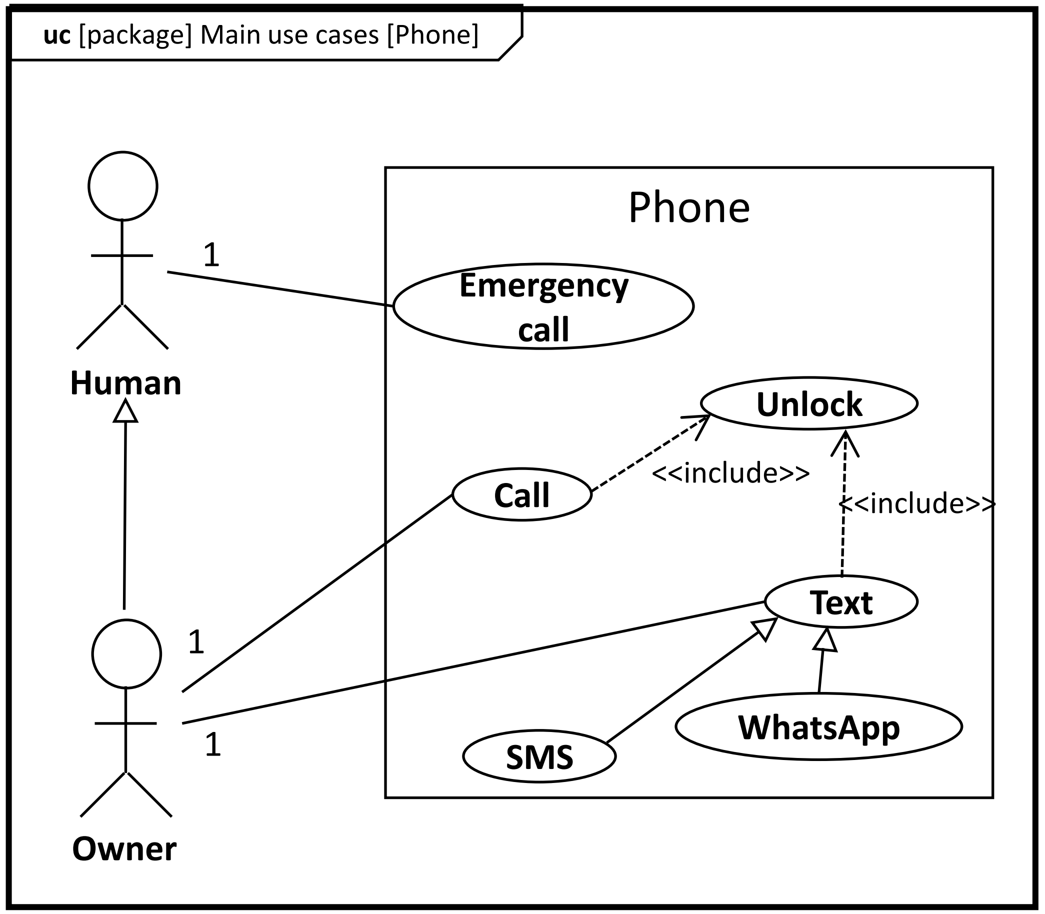
- generalization : a specialized actor participates in all use cases of general actor

use case relationships

- generalization : specialized use case is involved with the same actors/scenarios as the general use case
- inclusion: the included use case is always performed as part of the base use case

relation between actors and use cases

- related to use cases by bidirectional communication paths (associations)
- multiplicities on both ends, e.g., 1..* (default is 0..1)



multiplicity at actor end: numbers of actors involved in the use case

multiplicity at use-case end: number of instances in which the actor(s) can be involved at any one time

# SysML – use cases – construction

identifying use cases

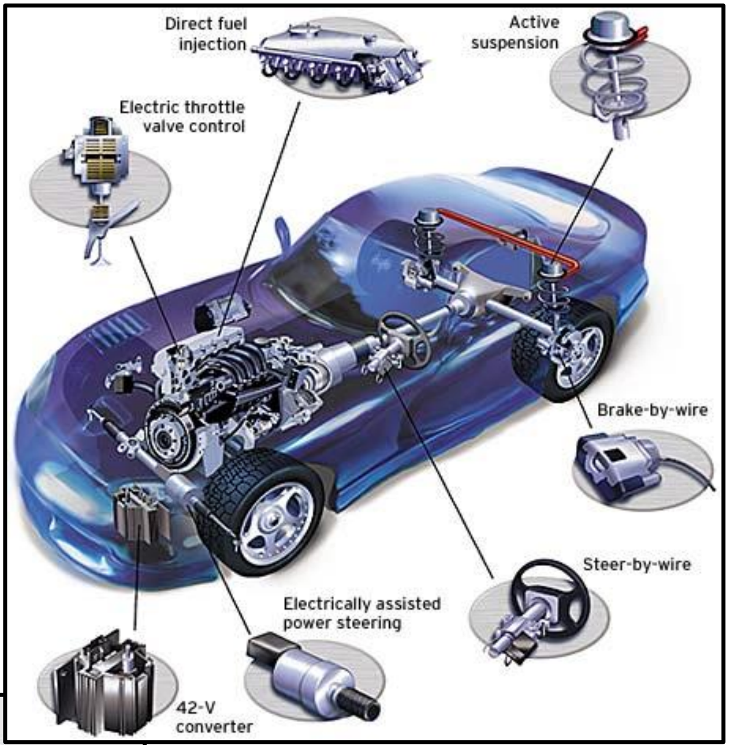
- different interactions, functions
  - what do users expect: the happy flow
  - normal operation, maintenance, testing, alternative executions / timings, initialization, …
  - failures, exceptions, time out, invalid input, oversize luggage, flooding, …
- ask the stakeholders!

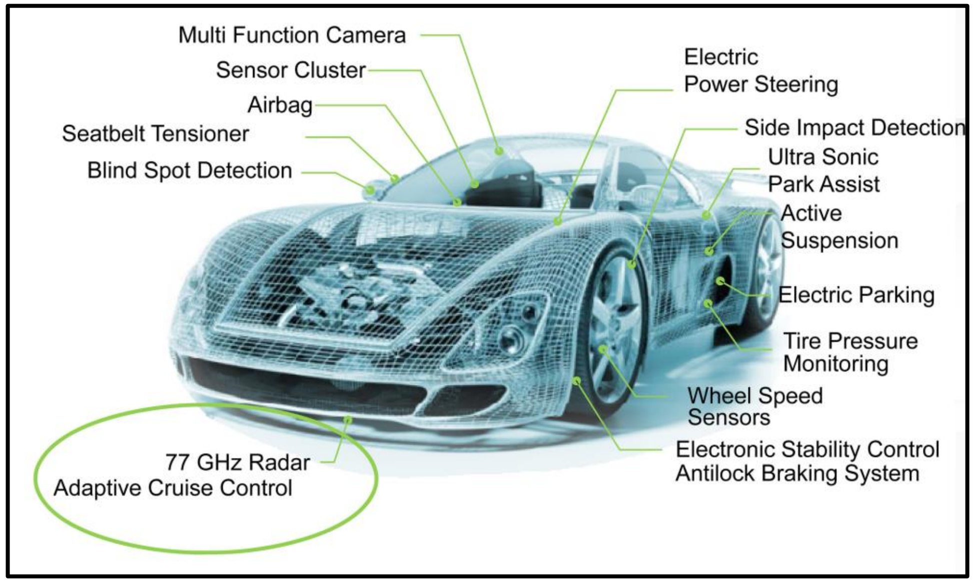
identifying actors

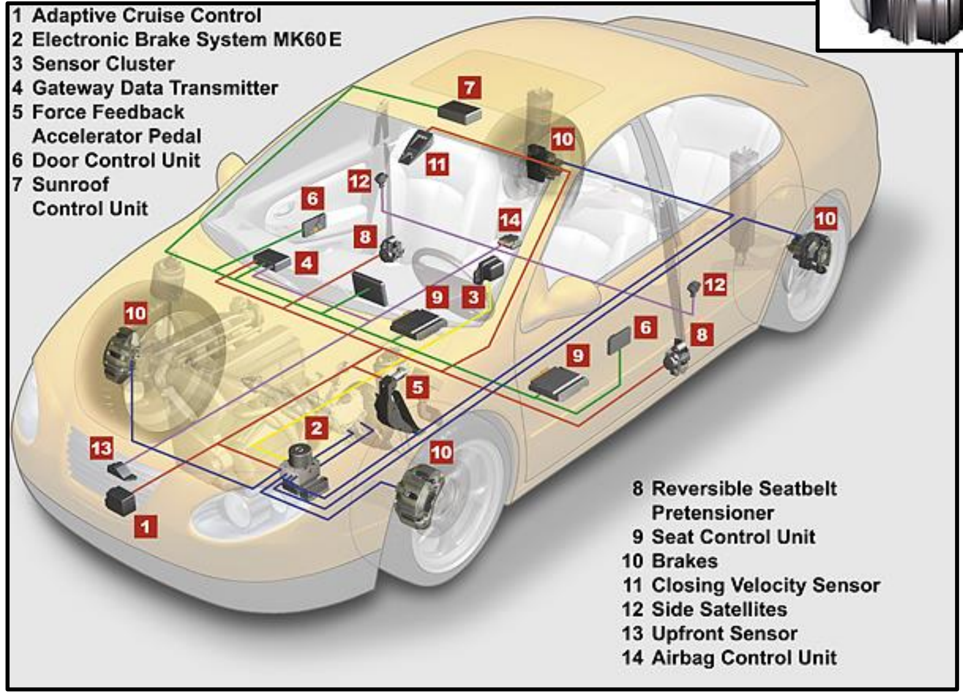
1. roles played by humans in the application domain
   - e.g., user, test engineer, operator, manager, employee, visitor, government
2. (sub)systems in the environment that communicate with the system
   - e.g., sensors, detectors, computers, networks
3. external factors that can produce events
   - e.g., overheating, start time, reset, overflow, luggage stuck on belt

## SysML – running example

We consider an automobile system. Vehicle occupants can operate the vehicle in several ways. They can enter and exit the vehicle, which involves opening a door. They can control the airco, and the entertainment system. The driver can drive the vehicle.







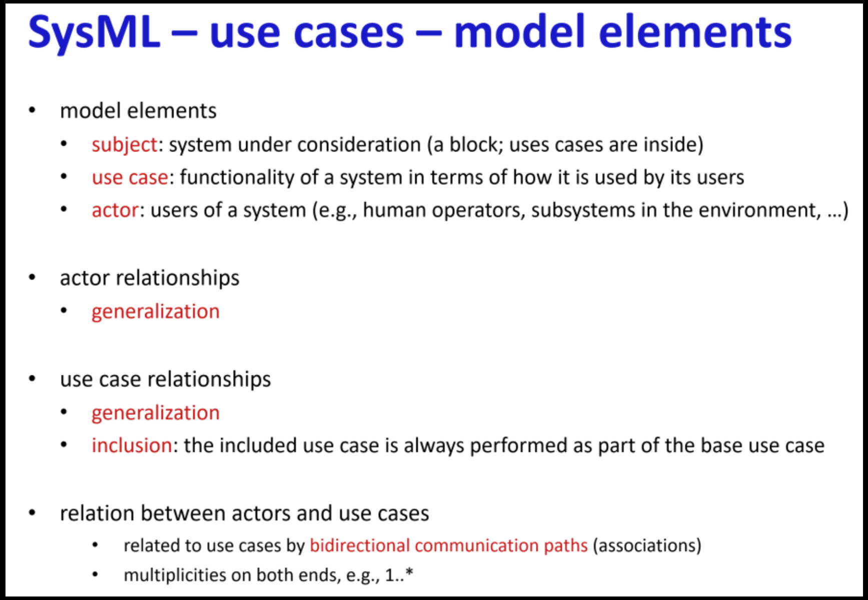
1 Adaptive Cruise Control
2 Electronic Brake System MK60E
3 Sensor Cluster
4 Gateway Data Transmitter
5 Force Feedback Accelerator Pedal
6 Door Control Unit
7 Sunroof Control Unit
8 Reversible Seatbelt Pretensioner
9 Seat Control Unit
10 Brakes
11 Closing Velocity Sensor
12 Side Satellites
13 Upfront Sensor
14 Airbag Control Unit

sources: motorola, aa1car.com

## think – pair – share

- what is the subject
- what are the use cases for Vehicle Operation
- what are the actors involved in Vehicle Operation
- what are the relations

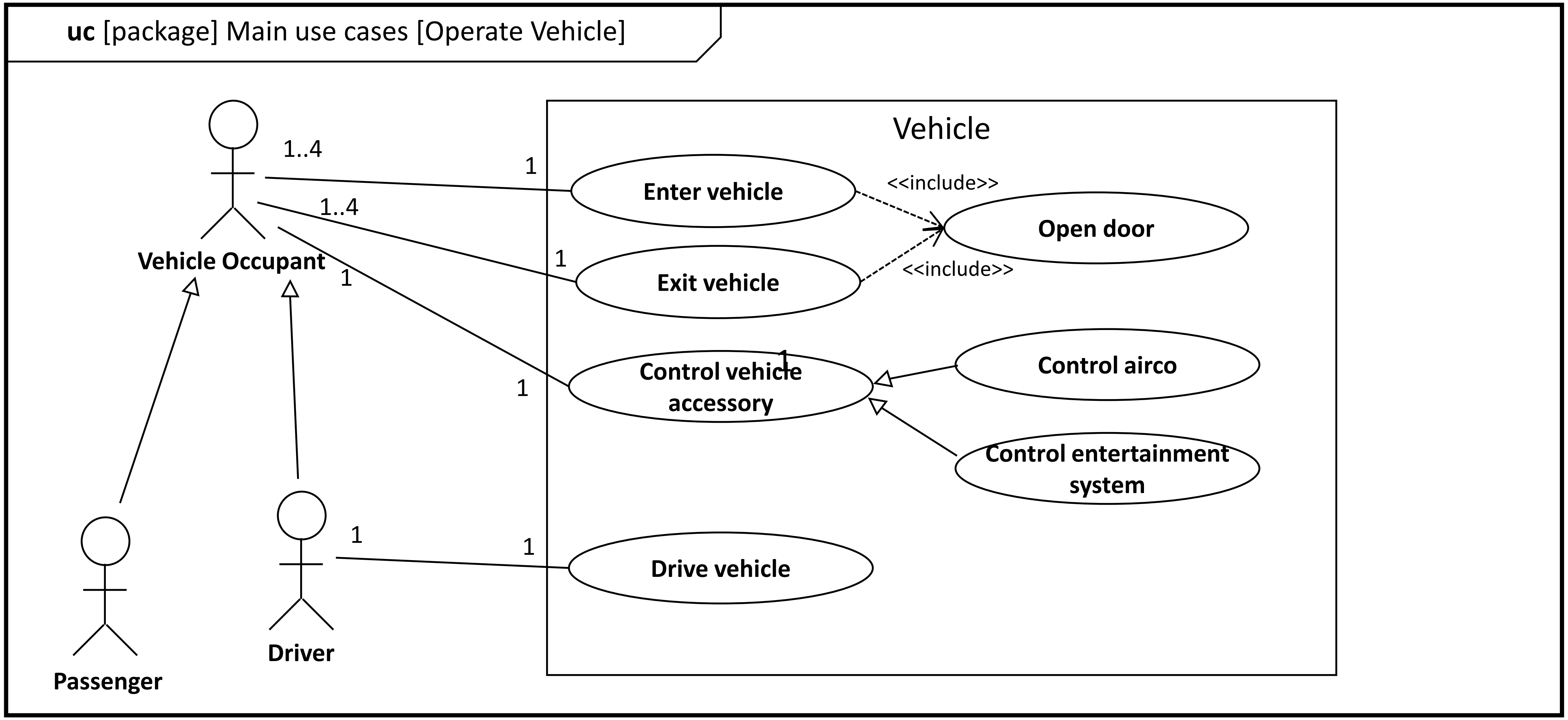
We consider an automobile system. Vehicle occupants can operate the vehicle in several ways. They can enter and exit the vehicle, which involves opening a door. They can control the airco, and the entertainment system. The driver can drive the vehicle.



### SysML – use cases – model elements

- model elements
  - subject: system under consideration (a block; uses cases are inside)
  - use case: functionality of a system in terms of how it is used by its users
  - actor: users of a system (e.g., human operators, subsystems in the environment, …)
- actor relationships
  - generalization
- use case relationships
  - generalization
  - inclusion: the included use case is always performed as part of the base use case
- relation between actors and use cases
  - related to use cases by bidirectional communication paths (associations)
  - multiplicities on both ends, e.g., 1..*

# SysML – use case diagram (uc)



## SysML – use cases – descriptions

the use case diagram by itself gives very little information; it must be accompanied by a use-case description that includes:

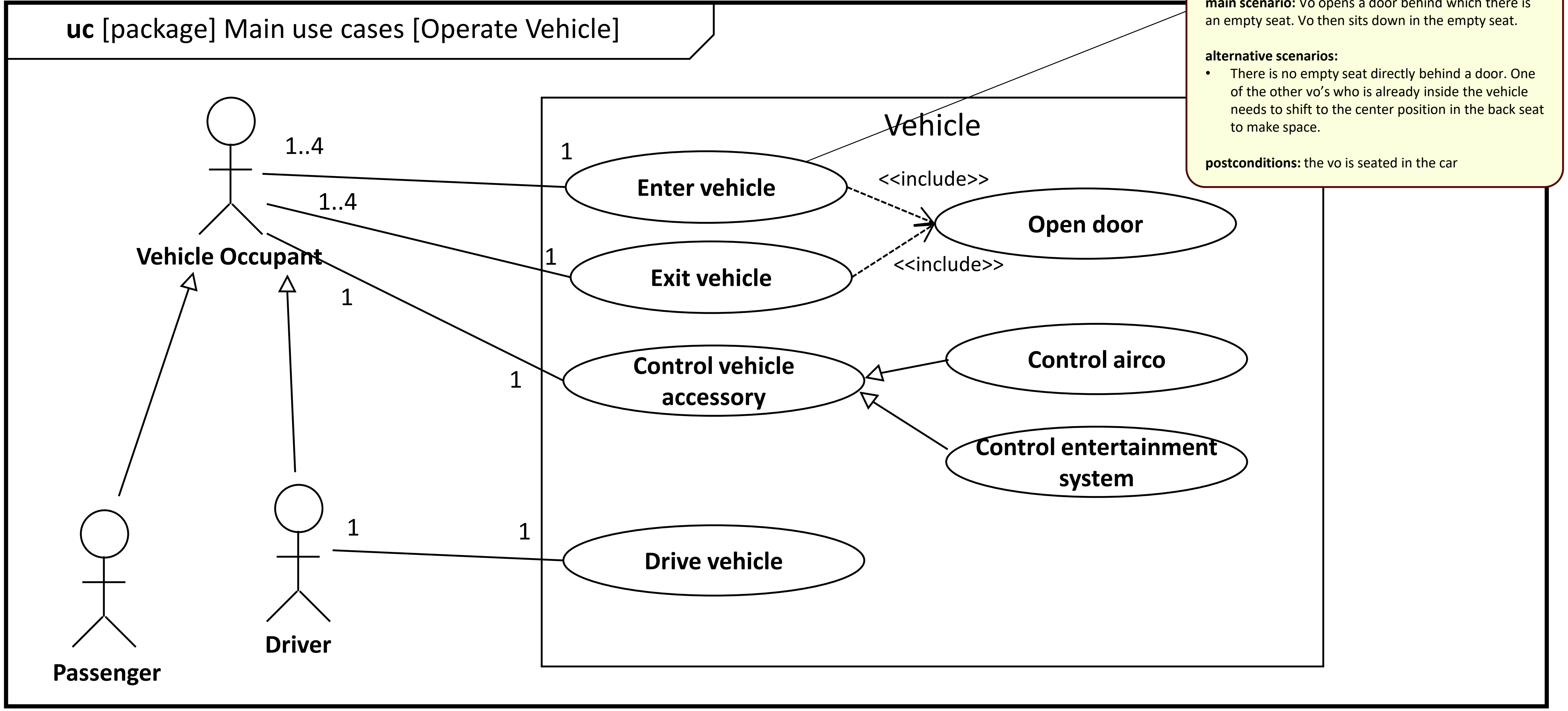
- **name**
- **primary actor**: who triggers the use case
- **supporting / secondary actors**: other participating actors
- **preconditions**: the conditions that must hold such that the use case can begin
- **main scenario**: what happens in this use case
- **alternative scenarios**:
  - list of related use cases, e.g. those that are less frequent or not normal
- **postconditions**: the conditions that must hold after the use case finishes

## SysML – use cases – descriptions

the use case diagram by itself gives very little information; it must be accompanied by a use-case description that includes:

- **name:** Enter vehicle
- **primary actor:** vehicle occupant (vo)
- **supporting / secondary actors:** vo already in the vehicle
- **preconditions:** there is a free seat in the car
- **main scenario:** Vo opens a door behind which there is an empty seat. Vo then sits down in the empty seat.
- **alternative scenarios:**
  - There is no empty seat directly behind a door. One of the other vo's who is already inside the vehicle needs to shift to the center position in the back seat to make space.
- **postconditions:** the vo is seated in the car

# SysML – use case diagram (uc)



**name:** Enter vehicle

**primary actor:** vehicle occupant (vo)

**supporting / secondary actors:** vo already in the vehicle

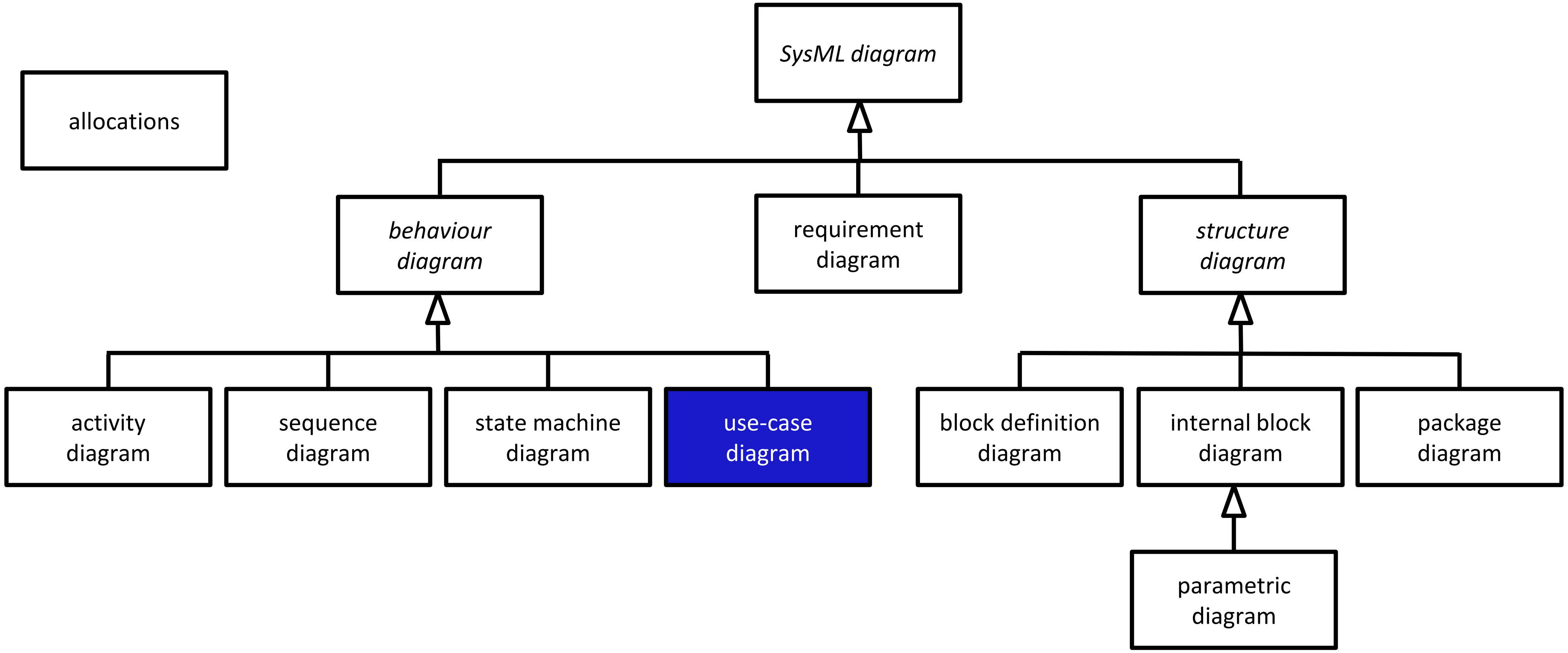
**preconditions:** there is a free seat in the car

**main scenario:** Vo opens a door behind which there is an empty seat. Vo then sits down in the empty seat.

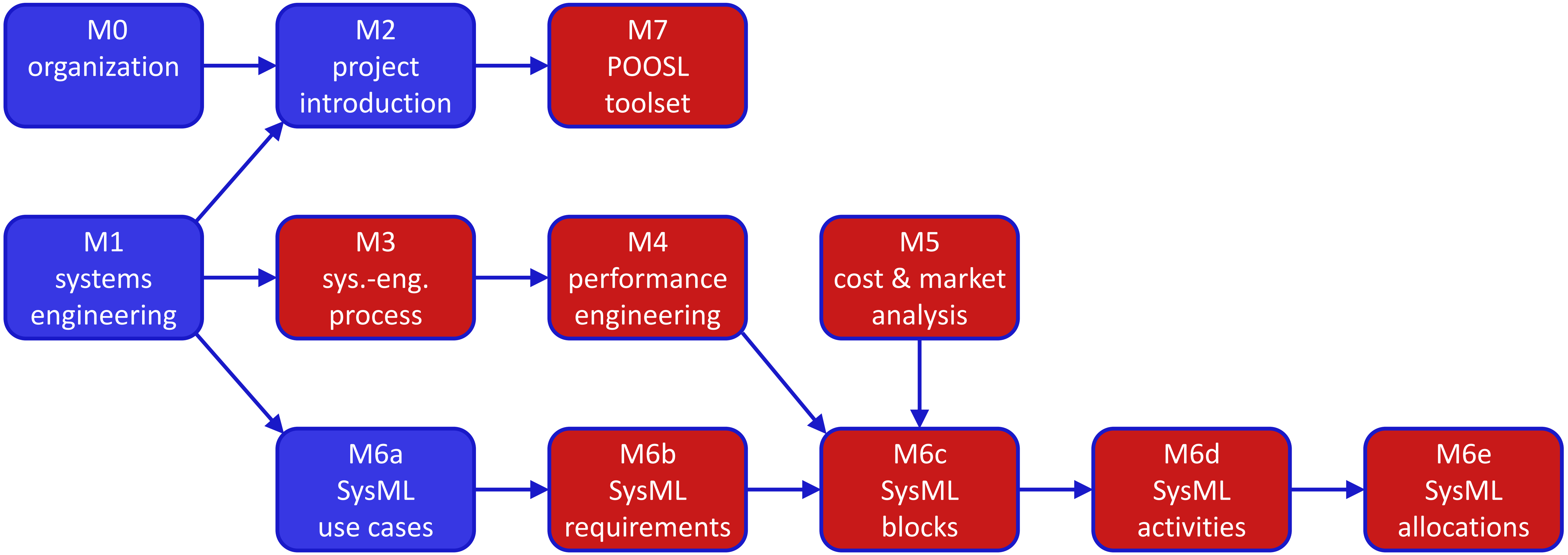
**alternative scenarios:**
- There is no empty seat directly behind a door. One of the other vo's who is already inside the vehicle needs to shift to the center position in the back seat to make space.

**postconditions:** the vo is seated in the car

# SysML – use cases



## modules


M0
organization
M2
project
introduction
M7
POOSL
toolset
M1
systems
engineering
M3
sys.-eng.
process
M4
performance
engineering
M5
cost & market
analysis
M6a
SysML
use cases
M6b
SysML
requirements
M6c
SysML
blocks
M6d
SysML
activities
M6e
SysML
allocations

## to remember

a SysML model is a collection of *model elements* (including their relations)

a SysML diagram is a *view* on the model

use cases model the top-level goals that the system is intended to support (user perspective)

textual use-case descriptions provide more information

SysML 1.6 spec for details/examples; especially Annex D (Sample problem)
https://www.omg.org/spec/SysML/1.6/PDF

## todo before next meeting

- install the tooling (see the project description)
  - Eclipse Papyrus
  - Eclipse POOSL
  - Eclipse TRACE4CPS
  - scripting environment of choice
- form groups of three or four
  - register groups in Canvas
- watch Papyrus videos on packages and use cases